1. Create the following table and enter at least 5 records in each.

```
create table manager(
managerid number(10) primary key,
fname varchar(30),
mname varchar(30),
lname varchar(20),
address varchar(30),
city varchar(40),
department varchar(40) check(department in('account','sales','marketing')),
gender char(1) check(gender in ('m','f')),
mobile_number number(10)
);

insert into manager values(11,'joker','gotam','city','hevan','swarg','account','m',9292929292);
insert into manager values(12,'harly','gotam','city','hell','nark','account','f',9000000000);
insert into manager values(13,'prince','dipakbhai','hirapara','hevan','swarg','account','m',9898989898);
insert into manager values(14,'norra','tridev','fatehi','hell','nark','sales','f',6969696969);
insert into manager values(15,'sunny','chote','leon','mumbai','bombey','marketing','f',6969686969);

create table employ(
managerid number(10) references manager(managerid),
employid number(20),
employee_name varchar(20),
department varchar(40)
);
insert into employ values(11,0011,'joker','account');
insert into employ values(12,0012,'harly','account');
insert into employ values(13,0013,'prince','account');
```

```
insert into employ values(14,0014,'noora','sales');

insert into employ values(15,0015,'sunny','marketing');
```

2)

```
declare

v_managerid manager.managerid %type;

v_fname manager.fname%type;

v_mname manager.mname%type;

v_lname manager.lname%type;

v_address manager.address%type;

v_city manager.city%type;

v_department manager.department%type;

v_gender manager.gender %type;

v_mobileno manager.mobile_number%type;

begin

v_managerid := :managerid;

v_fname:= :fname;

v_mname:= :mname;

v_lname:= :lname;

v_address:= :address;

v_city:= :city;

v_department:= :department;

v_gender:= :gender;

v_mobileno:= :mobileno;

insert into Manager values(v_managerid
,v_fname,v_mname,v_lname,v_address,v_city,v_department,v_gender,v_mobileno);
```

```
dbms_output.put_line('data inserted');
end;
```

```
data inserted

Statement processed.
```

0.09 seconds

3.

```
create or replace function totalemp2
return number is
total number(2);
begin
select count(*) into total from employ;
return total;
end;

declare
c number(2);
begin
c := totalemp();
dbms_output.put_line('total number of record is '||c);
```

Results Explain Describe Saved SQL History

```
Function created.
```

0.00 seconds

```
declare
c number(2);
begin
c := totalemp2();
dbms_output.put_line('total number of record is '||c);
end;
```

Results Explain Describe Saved SQL History

```
total number of record is 5

Statement processed.
```

0.28 seconds

```
select * from employ;
```

Results Explain Describe Saved SQL History

MANAGERID	EMPLOYID	EMPLOYEE_NAME	DEPARTMENT
11	11	joker	account
12	12	harly	account
13	13	prince	account
14	14	noora	sales
15	15	sunny	marketing

5 rows returned in 0.00 seconds CSV Export

2. Create a trigger which is executed on updatingemployeeidand employeenameinEmployeetable and the old row of data will be copied in backup table.

```
create or replace trigger emp_backup
before update or delete
on employ for each row
begin
insert into backup values (:OLD.employid ,:OLD.employee_name);
end;
create table backup as select *from EMPLOYEE_2 where 2=1;

drop table backup;

create table backup
```

Results Explain Describe Saved SQL History

```
Trigger created.
```

0.14 seconds